(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 211 600 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**05.06.2002 Bulletin 2002/23**

(51) Int Cl.[7]: **G06F 9/50**

(21) Application number: **01310045.8**

(22) Date of filing: **30.11.2001**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **04.12.2000 US 729371**

(71) Applicant: **International Business Machines Corporation**
**Armonk, NY 10504 (US)**

(72) Inventors:
- **Dillenberger, Donna,**
  **c/o IBM United Kingdom Ltd.**
  **Winchester, Hampshire SO21 2JN (GB)**
- **Hulber, Mark Francis,**
  **c/o IBM United Kingdom Ltd.**
  **Winchester, Hampshire SO21 2JN (GB)**

(74) Representative: **Burt, Roger James, Dr.**
**IBM United Kingdom Limited**
**Intellectual Property Department**
**Hursley Park**
**Winchester Hampshire SO21 2JN (GB)**

## (54) Policy management for distributed computing and a method for aging statistics

(57) A policy management system and method having a plurality of cooperating computers connected in a network. A policy management software resident in one or more managing computers of the network monitors the network and collects performance related values, such as, response time or queue delay of the cooperating computers. Performance related metrics are derived from the performance values and posted for access by the software that distributes work or controls execution of the work. The performance metrics includes only a number and average of the values received. The performance related values are formed in a data structure having n rows, where n is the number of value reporting intervals for which the performance metrics are kept. As a new current interval begins, the performance metrics of the nth row of a preceding interval are discarded and such nth row is used as the first row for the current interval. The remaining rows are shifted down one row position.

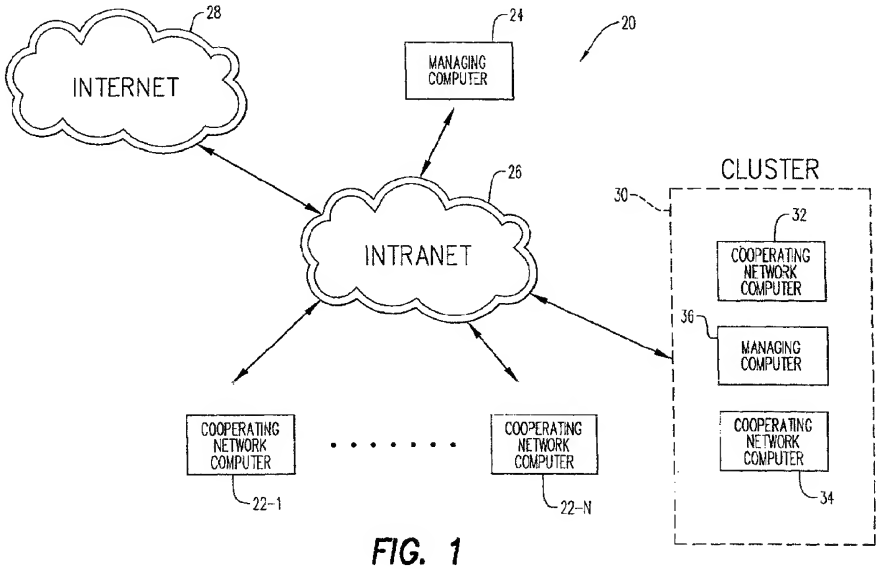


FIG. 1

EP 1 211 600 A2

## Description

**[0001]** This invention relates to a system and method for managing policy decisions for a network of cooperating computers and to a method for aging statistics.

**[0002]** Prior art methods that implement policy for computer systems that are homogenous in architecture assigned work on a platform specific basis or an operating system basis. For example, the IBM S/390 workload manager uses 390 system platform specific statistics, such as, multiprogramming level, virtual storage, expanded storage, to make decisions on where to place work. Prior art methods of policy management have required advance knowledge of how much CPU time or memory an application needs to run to efficiently assign the application in a cluster of computers and take advantage of the cluster resources. Prior art methods of policy management also have created an affinity between certain types of work and a specific computer.

**[0003]** Statistics used by prior art policy management methods have been updated on a periodic basis, whether or not new values have been received. This updating has been scheduled by a timer that signals the update times. This causes additional path length, CPU cycles and concerns about recovery, such as, failure of the timer to signal.

**[0004]** With the advent of computer networks and the distribution of work among the computers, there is a need for a policy manager that can manage policy independent of the architecture of the computers connected in the network.

**[0005]** There is also a need for a method and system for aging statistics that are used in the policy management process.

**[0006]** The present invention accordingly provides, in a first aspect, a method of managing the availability to do work of a plurality of cooperating computers connected in a network, said method comprising: (a) identifying a set of specific ones of said plurality of cooperating computers as available resources for the performance of said work; (b) receiving performance related values of said plurality of cooperating computers; (c) deriving performance related metrics from said performance related values; and (d) changing said set of specific ones of said plurality of cooperating computers based on said performance related metrics.

**[0007]** Preferably, at least one of said plurality of cooperating computers is heterogeneous with respect to the other cooperating computers of said plurality of cooperating computers.

**[0008]** Preferably, step (d) adds additional ones of said plurality of cooperating computers to said set or deletes one or more of said specific ones of said plurality of cooperating computers from said set.

**[0009]** Preferably, step (d) changes said set independently of any architecture or operating system specific metrics of said plurality of cooperating computers.

**[0010]** Preferably, step (d) changes said set independently of any workload specific metrics of said plurality of cooperating computers.

**[0011]** Preferably, said performance values are selected from the group consisting of: response times and queue delays.

**[0012]** Preferably, a cluster of said plurality of cooperating computers is connected to a node contained in said network, further comprising (e) requesting a manager of said cluster to accept additional work or to give up pending work based on said performance related metrics.

**[0013]** Preferably, the method further comprises (f) requesting said manager of said cluster to start more work or to run more pieces of an application on one or more of the cooperating computers of said cluster.

**[0014]** Preferably, step (a) identifies said set at a first time based on said performance related metrics, and wherein step (d) changes said set at a second later time.

**[0015]** Preferably, step (d) is performed only when a new value has been received or a request has been made to view the data.

**[0016]** Preferably, step (d) forms said performance metrics as an aggregation of said values.

**[0017]** Preferably, step (d) forms said performance metrics for each of said plurality of said cooperating computers.

**[0018]** Preferably, step (d) is performed only when a new one of said values is received or a request to view the performance metric is received.

**[0019]** Preferably, step (b) receives said performance related values over a series of time intervals, and wherein step (c) derives said performance metrics for n periods, of which the performance metrics of the nth period thereof includes an aggregate of the performance metrics for a current interval plus n-1 of the preceding intervals.

**[0020]** Preferably, the performance metrics of the nth period of a preceding interval are discarded during a current interval.

**[0021]** Preferably, said performance metrics for each of said periods include only a number and average of values received.

**[0022]** Preferably, each of said performance metrics includes only a number and an average of values received.

**[0023]** Preferably, step (c) forms said performance metrics as a data structure having n rows that contain the performance metrics of said n periods, respectively, wherein the performance metrics of the nth row of a preceding interval are discarded during a current interval, and wherein said nth row of the preceding interval is used as a first row in the current interval and the remaining ones of said n rows are shifted down one row position.

**[0024]** In a second aspect, the present invention provides a computer having a CPU and a memory comprising:

policy program means for causing said CPU to mange the availability to do work of a plurality of cooperating computers that are connected in a network, said policy program means comprising:

first means for performing a first operation that identifies a set of specific ones of said plurality of cooperating computers as available resources for the performance of work;

second means for performing a second operation that receives performance related values of said plurality of cooperating computers;

third means for performing a third operation that derives performance related metrics from said performance related values; and

fourth means for performing a fourth operation that changes said set of specific ones of said plurality of cooperating computers based on said performance related metrics.

**[0025]** Preferably, at least one of said plurality of cooperating computers is heterogeneous with respect to the other cooperating computers of said plurality of cooperating computers.

**[0026]** Preferably, said fourth operation adds additional ones of said plurality of cooperating computers to said set or deletes one or more of said specific ones of said plurality of cooperating computers from said set.

**[0027]** Preferably, said fourth operation changes said set independently of any architecture or operating system specific metrics of said plurality of cooperating computers.

**[0028]** Preferably, said fourth operation changes said set independently of any workload specific metrics of said plurality of cooperating computers.

**[0029]** Preferably, said performance values are selected from the group consisting of: response times and queue delays.

**[0030]** Preferably, a cluster of said plurality of cooperating computers is connected to a node contained in said network, further comprising a fifth means for performing a fifth operation that requests a manager of said cluster to accept additional work or to give up pending work based on said performance related metrics.

**[0031]** Preferably, the computer of the second aspect further comprises sixth means for performing a sixth operation that requests said manager of said cluster to start more work or to run more pieces of an application on one or more of the cooperating computers of said cluster.

**[0032]** Preferably, said first operation identifies said set at a first time based on said performance related metrics, and wherein said fourth changes said set at a second later time.

**[0033]** Preferably, said fourth operation is performed only when a new value has been received or a request has been made to view the data.

**[0034]** Preferably, said fourth operation forms said performance metrics as an aggregation of said values.

**[0035]** Preferably, said fourth operation forms said performance metrics for each of said plurality of said cooperating computers.

**[0036]** Preferably, said fourth operation is performed only when a new one of said values is received or a request to view the performance metric is received.

**[0037]** Preferably, said second operation receives said performance related values over a series of time intervals, and wherein said third operation derives said performance metrics for n periods, of which the performance metrics of the nth period thereof includes an aggregate of the performance metrics for a current interval plus n-1 of the preceding intervals.

**[0038]** Preferably, the performance metrics of the nth period of a preceding interval are discarded during a current interval.

**[0039]** Preferably, said performance metrics for each of said periods include only a number and average of values received.

**[0040]** Preferably, each of said performance metrics includes only a number and an average of values received.

**[0041]** Preferably, said third operation forms said performance metrics as a data structure having n rows that contain the performance metrics of said n periods, respectively, wherein the performance metrics of the nth row of a preceding interval are discarded during a current interval, and wherein said nth row of the preceding interval is used as a first row in the current interval and the remaining ones of said n rows are shifted down one row position.

**[0042]** In a third aspect, the present invention comprises a computer program to, when loaded into a computer system and executed, cause said computer system to perform the steps of a method according to the first aspect. Preferred computer program features of the third aspect correspond to the preferred method features of the first aspect.

**[0043]** A preferred embodiment of the present invention addresses the aforementioned needs with a policy management system and method that manages the availability of a plurality of cooperating computers connected in a network to do work. The method identifies a set of specific ones of the plurality of cooperating computers as available resources for the performance of the work. Performance metrics are derived from performance related values of the plurality of cooperating computers. Based on the performance metrics, the set of specific ones of the cooperating computers are changed to thereby enable the allocation of work to cooperating computers that, from a performance standpoint, can do the work in the shortest possible time.

**[0044]** The system and method of the invention can be used with cooperating computers that are either ho-

mogenous or heterogeneous in architecture or that employ the same or diverse operating systems. The performance related metrics, for example, include response times or queue delays of a cooperating computer.

**[0045]** The system and method of policy management preferably is flexible to allow as much policy management as desired to be delegated from a system policy manager to a local or cluster manager. The method may request a manager of a cluster to accept additional work or to give up pending work based on the performance related metrics or to start more work or to run more pieces of an application on one or more of the cooperating computers of the cluster.

**[0046]** According to a preferred feature of the present invention, the performance metrics are derived or updated only when a new value has been received or a request has been made to view the data. This saves CPU cycles that were used in prior methods that performed updates periodically, whether or not new values had been received since the last update.

**[0047]** According to another preferred feature of the invention, the performance related values are received over a series of time intervals. The performance metrics are derived for n periods, of which the performance metrics of the nth period thereof includes an aggregate of the performance metrics for the current interval plus n-1 of the preceding time intervals. The performance metrics of the nth or last period of a preceding interval are discarded during a current interval.

**[0048]** The performance metrics are preferably formed as a data structure having n rows that contain the performance metrics of the n periods. The performance metrics of the nth row of a preceding interval are discarded during a current interval. The nth row of the preceding interval is used as a first row in the current interval and the remaining rows are shifted down one row position.

**[0049]** According to a further preferred feature of the present invention, each of the performance metrics includes only a number and average of values received.

**[0050]** A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 is a block diagram of a system in which the policy management system and method of a preferred embodiment of the present invention is employed;

FIG. 2 is a block diagram of the managing computer of the FIG. 1 system;

FIG. 3 is a flow diagram of the policy manager program of the FIG. 2 managing computer;

FIG. 4 is a flow diagram of the performance values and metrics formation process of the FIG. 2 managing computer;

FIG. 5 depicts a data structure of the performance metrics formed by the process of FIG. 4;

FIG. 6 is a flow diagram of the receive new values portion of the flow diagram of FIG. 4; and

FIG. 7 is a flow diagram of the update table portion of the flow diagram of FIG. 4.

**[0051]** Referring to FIG. 1, a distributed computing system 20 includes a plurality of cooperating network computers 22-1 through 22-N, a managing computer 24, an intranet 26, an internet 28 and a cluster 30. Intranet 26 interconnects cooperating network computers 22-1 through 22-N, cluster 30 and managing computer 24 in a distributing computing network. Intranet 26 is generally a network that is internal to an organization. A preferred embodiment of the present invention contemplates that other computers outside the organization may be included in the distributed computing network. To this end, internet 28 serves to interconnect these other computers (not shown) with managing computer 24, cluster 30 and cooperating network computers 22-1 through 22-N. Other non-network computers (not shown) can also communicate with system 20 via internet 28.

**[0052]** Cluster 30 includes a managing computer 36 and two cooperating network computers 32 and 34. Cooperating network computers 32 and 34 and managing computer 36 are each connected with intranet 26. Cluster 30 may be considered a node in system 20. Although two cooperating network computers 32 and 34 are shown, it will be apparent to those skilled in the art that cluster 30 may include more or less cooperating computers. It will also be apparent to those skilled in the art that system 20 may include additional clusters.

**[0053]** Managing computer 24 manages the policy concerning allocation of work (applications) and distribution of that work among cooperating computers 22-1 through 22-N and cluster 30. Managing computer 36 manages the policy and distribution of work among cooperating network computers 32 and 34. It will be apparent to those skilled in the art that managing computers 24 and 26 are shown as single computers by way of example and that policy management and work distribution may be functionally distributed among two or more computers.

**[0054]** Referring to FIG. 2, managing computer 24 includes a central processing unit (CPU) 40, an input/output (I/O) units section 42, a communications unit 44 and a memory 46. Communications unit 44 serves to interconnect managing computer with intranet 26 for communication with cooperating computers 22-1 through 22-N and cluster 30. Memory 46 includes an operating system program 48, a distribution manager program 52, a policy manager program 54, a metrics program 56 and

a metrics data structure 58.

**[0055]** Policy manager program 54 and metrics program 56 control CPU 40 to develop and update metrics data structure 58 with performance related metrics of cooperating network computers 32, 34 and 22-1 through 22-N. Distribution manager 52 uses metrics data structure 58 to control CPU 40 to distribute work among cooperating network computers 32, 34 and 22-1 through 22-N. Dependent on the allocation of policy management responsibility among managing computer 24 and managing computer 36, managing computer 36 may also have access to metrics data structure 58. It will be apparent to those skilled in the art that although metrics program 56 and metrics data structure 58 are shown as separate modules, each could be incorporated into policy manager program 54. It will be apparent to those skilled in the art that managing computer 36 may have the same or similar structure and software as managing computer 24.

**[0056]** Software stored in memory 46, including policy manager program 54, metrics program 56 may be loaded or down loaded from a memory medium 60 to memory 46.

**[0057]** Referring to FIG. 3, policy manager program 54 at step 70 identifies a set of resources that are available for work. For example, the specific ones of cooperating network computers 32, 34 or 22-1 through 22-N that are available for work are identified. This identification is available for access by distribution manager program 52. Step 72 examines the state of system 20 including metrics data structure 58. Step 74 determines if there is any need to change the available resources based on the metrics data structure 58. If no change is needed, step 76 causes policy manager program 54 to wait and then step 72 is repeated. If step 74 determines there is a need for change, step 78 either increases or decreases the current set of available resources. After step 78 is completed, step 76 causes policy manager program 54 to wait and then step 72 is repeated.

**[0058]** Referring to FIG. 4, metrics program 56 starts with step 80 that initializes based on provided parameters, such as, the location of basic components. Step 82 then causes metrics program 56 to wait until a request is received. If the request is for a report, the metrics contained in metric data structure 58 are updated by step 86. Step 88 then issues the requested report and control returns to step 82. If the request is to provide new values, step 90 derives the metrics from the new values. The derived metrics are then used by step 92 to update metric data structure 58. Control is then passed to step 82. Metrics program 56 performs the foregoing steps for each cooperating network computer that is connected in system 20, unless managing computer 36 is responsible to track the performance related values of cooperating network computers 32 and 34.

**[0059]** Referring to FIG. 5, metrics data structure 58 is shown in the form of a table having rows 100, 102, 104 and 106. Rows 100, 102, 104 and 106 each contain a performance related metric for a different value reporting interval. According to one aspect of the invention, the performance related value may be either the response time or queue delay of a cooperating network computer. In FIG. 5, the performance related value is shown as response time. The derived metrics in each row consist of a number of values received for that reporting interval and an average of the reported values. Thus, row 100 is shown with a total number of 7 values received thus far in a current reporting interval with an average response time of 160 milliseconds (ms). Thus, only two metrics are needed for each reporting interval. Row 102 has a reporting interval that consists of the previous interval plus the current interval. Row 104 has a reporting interval of the two previous intervals plus the current interval. Row 206 has a reporting interval of the three preceding intervals plus the current interval. It will be apparent to those skilled in the art that there can be more or less rows than the four rows 100, 102, 104 and 106.

**[0060]** Referring to FIG. 6, aggregate values step 90 of FIG. 4 is shown in detail. Step 110 waits for a new value. Step 112 receives a new value. Step 114 determines if the current value reporting interval has expired. If so, step 116 updates rows 100, 102, 104 and 106 of metrics data structure 58 and passes control to step 110. If not, step 118 updates only the current row 100. By updating only the current row as each new value is received, computation time is conserved. Step 120 increments the values number and step 122 calculates a new average. For example, the new average is the old average plus the difference of the current value and the old average divided by the new number of values. After step 122, control is passed to step 110.

**[0061]** Referring to FIG. 7, update table rows step 116 of FIG. 6 is shown in detail. Step 130 rotates table rows 100, 102, 104 and 106 of metrics data structure 58 of FIG. 5. Step 132 discards the metrics contained in the nth row, which is row 106 in FIG. 5. Row 106 is then used as the first row for the metrics to be derived in the next current interval. Step 134 then shifts all remaining rows down one position. Thus, for the next current interval, row 106 will contain the metrics for the current interval. Row 100 will contain the metrics for the previous interval plus that current interval. Row 102 will contain the metrics for the two previous intervals plus the current interval. Row 104 will contain the metrics for the three previous intervals plus the current interval.

**[0062]** Policy manager program 54 runs independently of distribution and execution of work within system 20. That is, the process of policy management is disconnected from the active work processes that are being managed. This has the effect of reducing overhead of system 20, thereby allowing for maximum scalability. Policy manager program 54 contains no process that establishes an affinity between a specific work item or type of work and a specific cooperating network computer.

**[0063]** Policy manager program 54 oversees system 20 and the work execution process without interfering with the production cycles of system 20. This is accomplished by having policy manager program 54 run separately from distribution manager program 52 and other software that controls execution of work by system 20. This allows policy manager program to monitor the execution of work and to extract decision making information, such as performance related values, while minimally impacting the ongoing work process.

**[0064]** Policy manager program 54 and managing computer 24 are a central policy manager that monitors system 20. In order to achieve fault tolerance as well as limit the load on any one policy manager, managing computer 24 can communicate with a plurality of local or cluster policy managers, such as, managing computer 36, throughout system 20. The task of each local managing computer is to enforce local policy, while meeting the goals of the global system through communication with other local policy managers and the central managing computer 24. Each cluster policy manager keeps track of the current policy and state of its cluster. By managing policy local to a cluster, the bottleneck of funneling all local policy decisions through a single central policy manager is avoided. Clusters, such as cluster 30, can be partitioned based on functionality, proximity or any arbitrary consideration.

**[0065]** The policy management system and method of a preferred embodiment of the present invention allows for more than one method for handling the distribution of policy management among managing computer 24 and cluster managing computers, such as, managing computer 36. In one aspect, managing computer 24 can view cluster 30 as a single node and have no knowledge of cooperating network computers 32 and 34. In another aspect of the invention, managing computer 24 must approve of all decisions made by managing computer 36 and, thus, has first hand knowledge of system 20. A combination of these two aspects allows the cluster managing computer 36 to make local decisions about its resource management, while central managing computer 24, as needed, has access to the state of cluster 30 and cooperating network computers 32 and 34.

## Claims

1. A method of managing the availability to do work of a plurality of cooperating computers connected in a network, said method comprising:

   (a) identifying a set of specific ones of said plurality of cooperating computers as available resources for the performance of said work;

   (b) receiving performance related values of said plurality of cooperating computers;

   (c) deriving performance related metrics from said performance related values; and

   (d) changing said set of specific ones of said plurality of cooperating computers based on said performance related metrics.

2. A method as claimed in claim 1, wherein at least one of said plurality of cooperating computers is heterogeneous with respect to the other cooperating computers of said plurality of cooperating computers.

3. A method as claimed in claim 1 or claim 2, wherein step (d) changes said set independently of at least one of:

   any architecture or operating system specific metrics of said plurality of cooperating computers; or

   any workload specific metrics of said plurality of cooperating computers.

4. A method as claimed in any of claims 1 to 3, wherein a cluster of said plurality of cooperating computers is connected to a node contained in said network, further comprising (e) requesting a manager of said cluster to accept additional work or to give up pending work based on said performance related metrics.

5. A method as claimed in claim 4, further comprising (f) requesting said manager of said cluster to start more work or to run more pieces of an application on one or more of the cooperating computers of said cluster.

6. A method as claimed in any preceding claim, wherein step (a) identifies said set at a first time based on said performance related metrics, and wherein step (d) changes said set at a second later time.

7. A method as claimed in any preceding claim, wherein step (d):

   forms said performance metrics as an aggregation of said values;
   forms said performance metrics for each of said plurality of said cooperating computers; and

   wherein step (d) is performed only when a new one of said values is received or a request to view the performance metric is received.

8. A method as claimed in any preceding claim, wherein step (b) receives said performance related values over a series of time intervals;

wherein step (c) derives said performance metrics for n periods, of which the performance metrics of the nth period thereof includes an aggregate of the performance metrics for a current interval plus n-1 of the preceding intervals;

wherein the performance metrics of the nth period of a preceding interval are discarded during a current interval; and

wherein step (c) forms said performance metrics as a data structure having n rows that contain the performance metrics of said n periods, respectively, wherein the performance metrics of the nth row of a preceding interval are discarded during a current interval, and wherein said nth row of the preceding interval is used as a first row in the current interval and the remaining ones of said n rows are shifted down one row position.

**9.** A computer having a CPU and a memory comprising:

policy program means for causing said CPU to mange the availability to do work of a plurality of cooperating computers that are connected in a network, said policy program means comprising:

first means for performing a first operation that identifies a set of specific ones of said plurality of cooperating computers as available resources for the performance of work;
second means for performing a second operation that receives performance related values of said plurality of cooperating computers;
third means for performing a third operation that derives performance related metrics from said performance related values; and
fourth means for performing a fourth operation that changes said set of specific ones of said plurality of cooperating computers based on said performance related metrics.

**10.** A computer program comprising computer program code to, when loaded into a computer system and executed, cause said computer system to perform the steps of a method as claimed in any of claims 1 to 8.

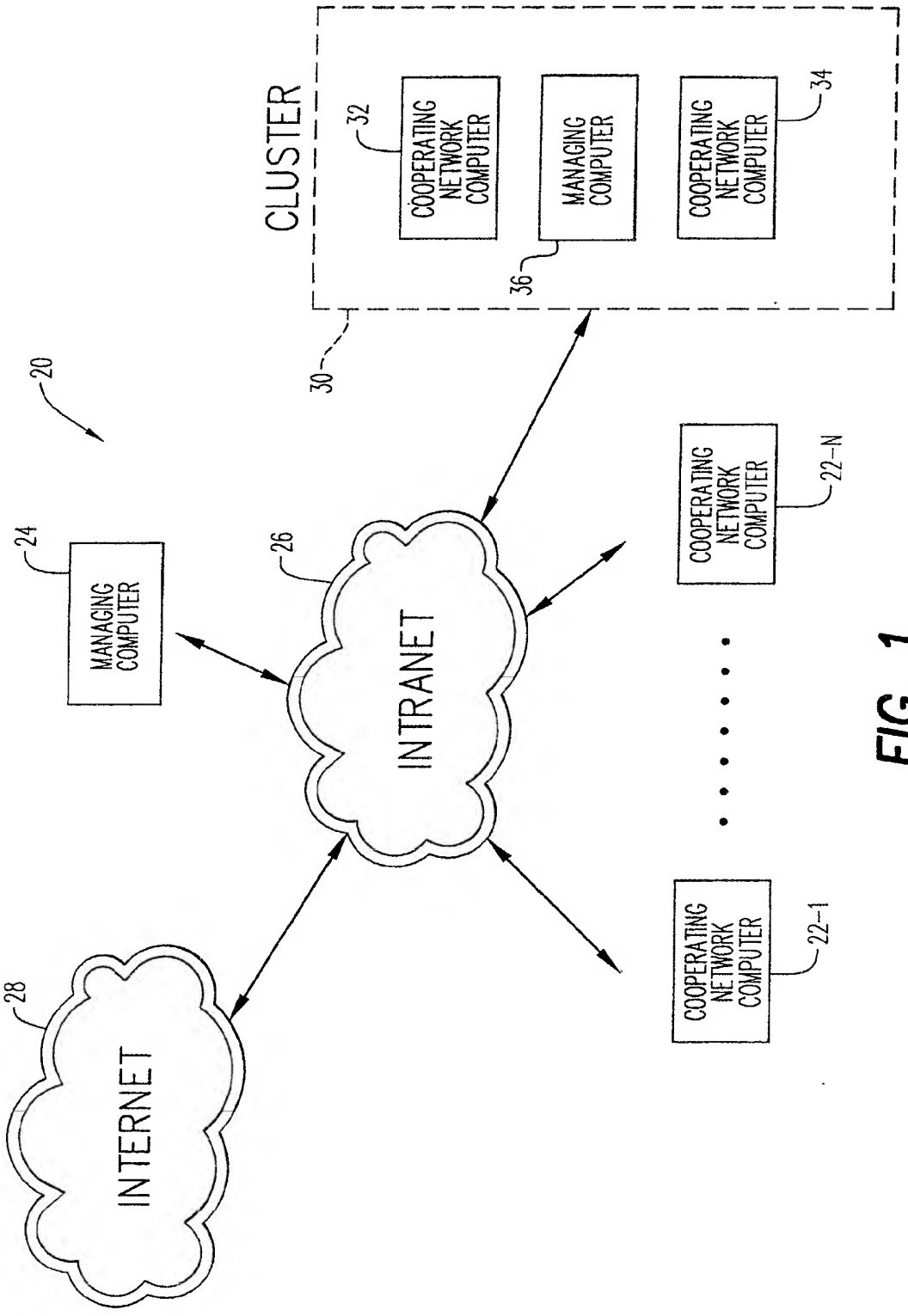
CLUSTER
32
COOPERATING NETWORK COMPUTER
MANAGING COMPUTER
COOPERATING NETWORK COMPUTER
34
36
30
20
24
MANAGING COMPUTER
26
INTRANET
COOPERATING NETWORK COMPUTER
22-N
COOPERATING NETWORK COMPUTER
22-1
28
INTERNET

FIG. 1

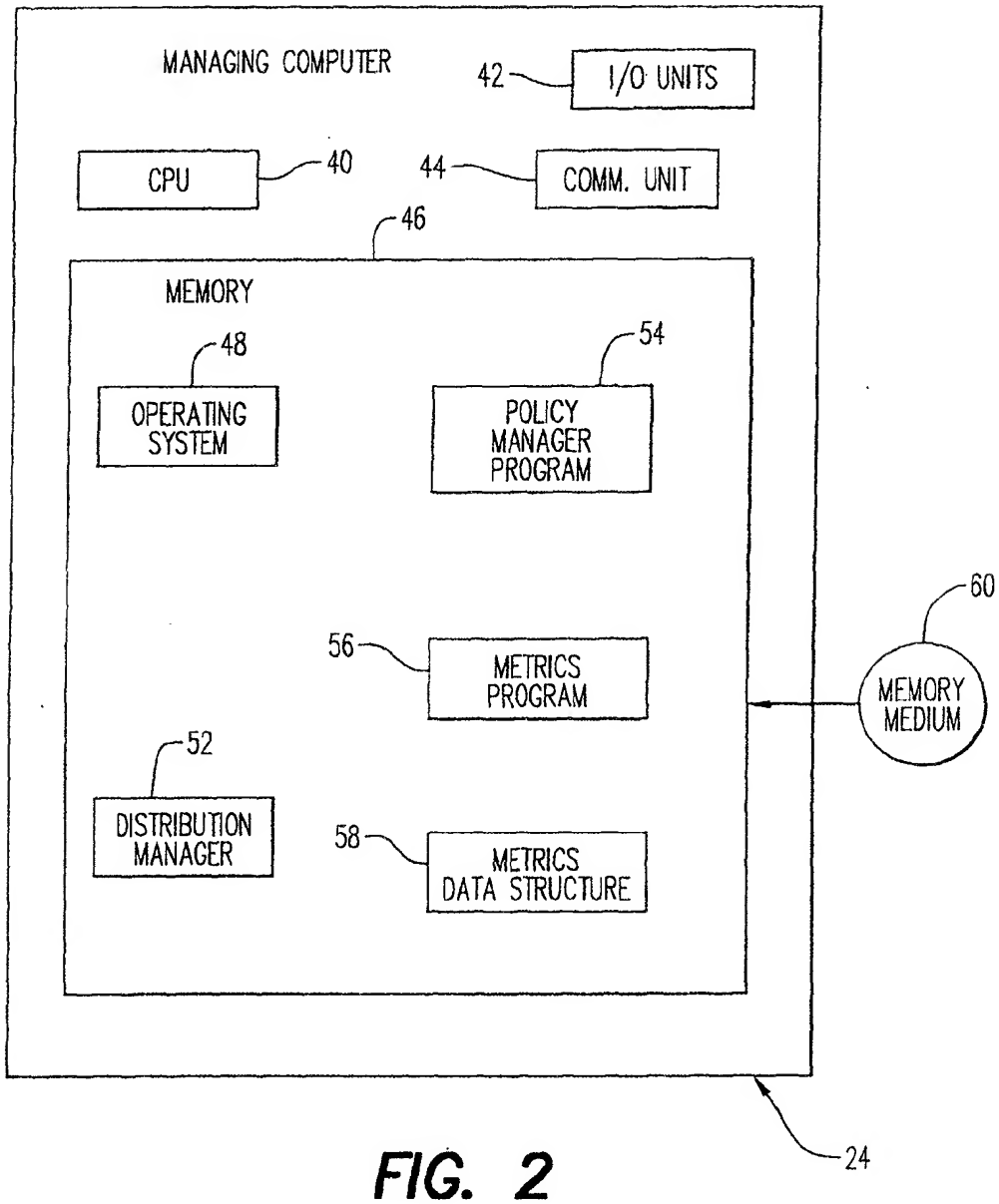
MANAGING COMPUTER
42
I/O UNITS
CPU
40
44
COMM. UNIT
46
MEMORY
48
OPERATING SYSTEM
54
POLICY MANAGER PROGRAM
60
56
METRICS PROGRAM
MEMORY MEDIUM
52
DISTRIBUTION MANAGER
58
METRICS DATA STRUCTURE
24

FIG. 2

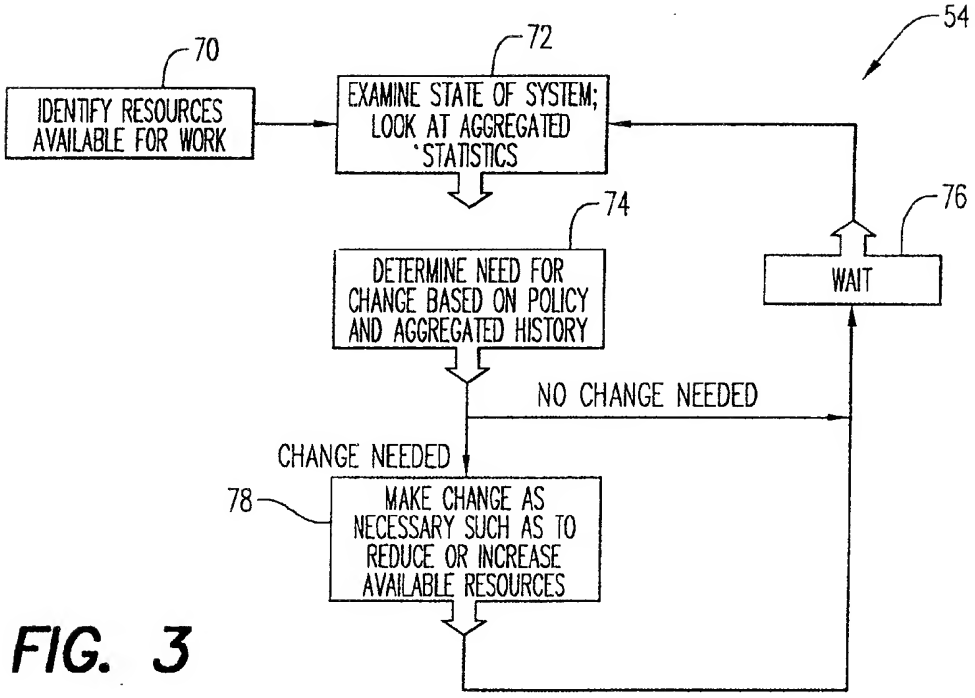
54
70
IDENTIFY RESOURCES AVAILABLE FOR WORK
72
EXAMINE STATE OF SYSTEM; LOOK AT AGGREGATED STATISTICS
74
DETERMINE NEED FOR CHANGE BASED ON POLICY AND AGGREGATED HISTORY
76
WAIT
NO CHANGE NEEDED
CHANGE NEEDED
78
MAKE CHANGE AS NECESSARY SUCH AS TO REDUCE OR INCREASE AVAILABLE RESOURCES

FIG. 3

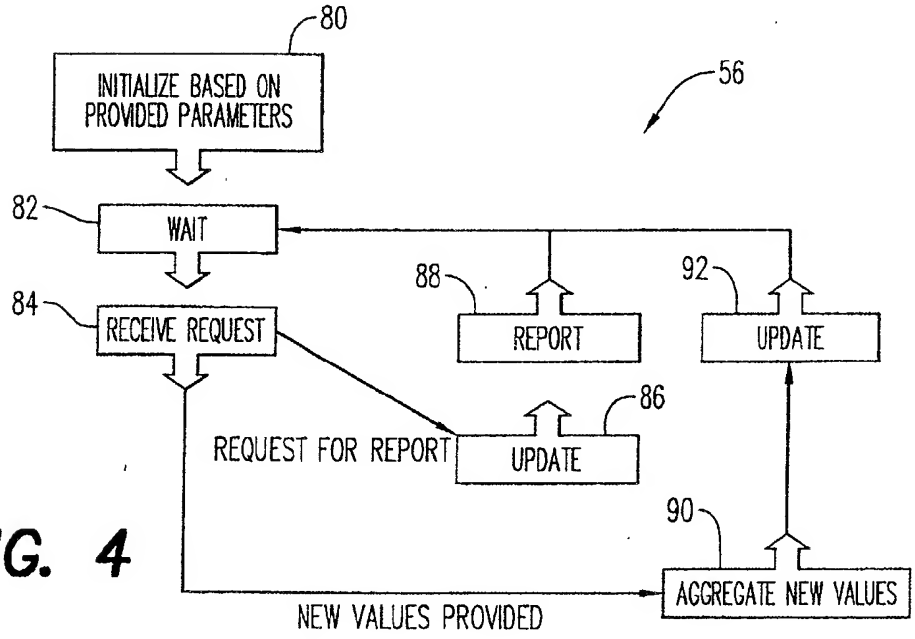
80
INITIALIZE BASED ON PROVIDED PARAMETERS
56
82
WAIT
84
RECEIVE REQUEST
88
REPORT
92
UPDATE
86
REQUEST FOR REPORT
UPDATE
90
AGGREGATE NEW VALUES
NEW VALUES PROVIDED

FIG. 4

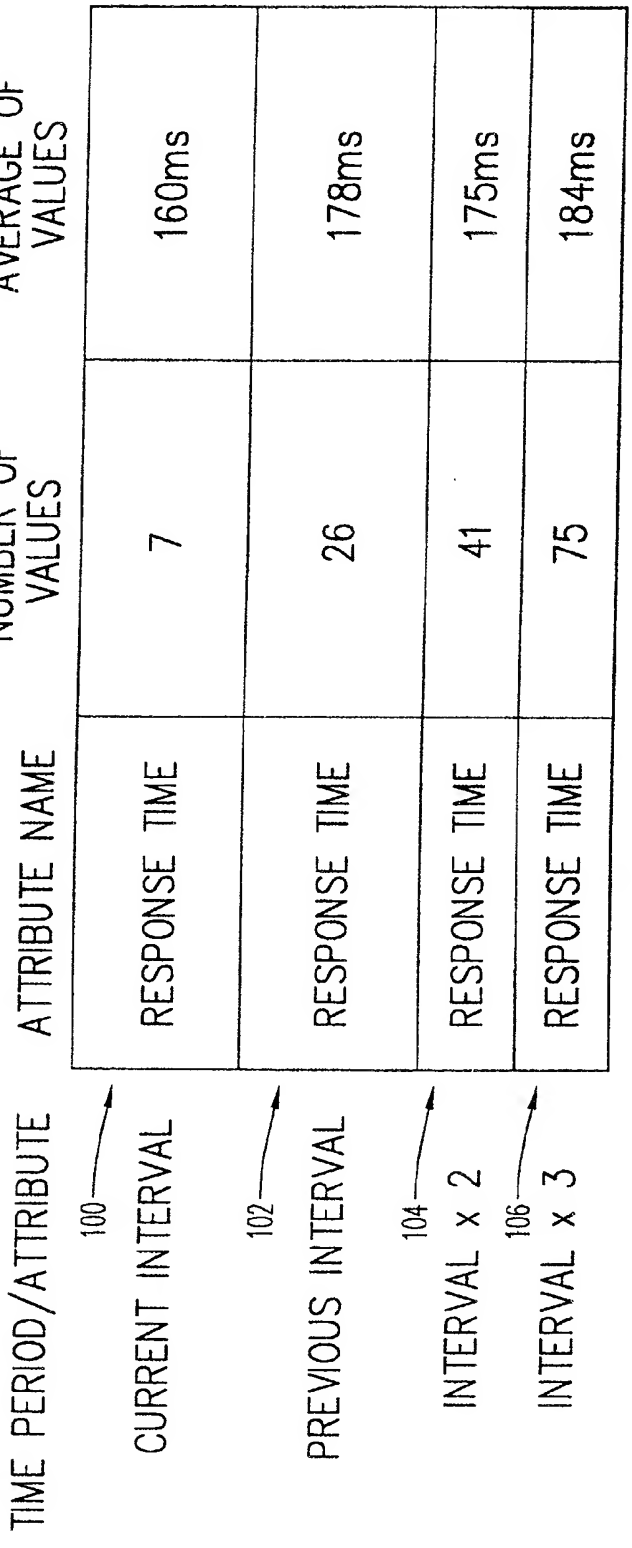
58
TIME PERIOD/ATTRIBUTE
ATTRIBUTE NAME
NUMBER OF VALUES
AVERAGE OF VALUES
100
CURRENT INTERVAL
RESPONSE TIME
7
160ms
102
PREVIOUS INTERVAL
RESPONSE TIME
26
178ms
104
INTERVAL x 2
RESPONSE TIME
41
175ms
106
INTERVAL x 3
RESPONSE TIME
75
184ms

FIG. 5

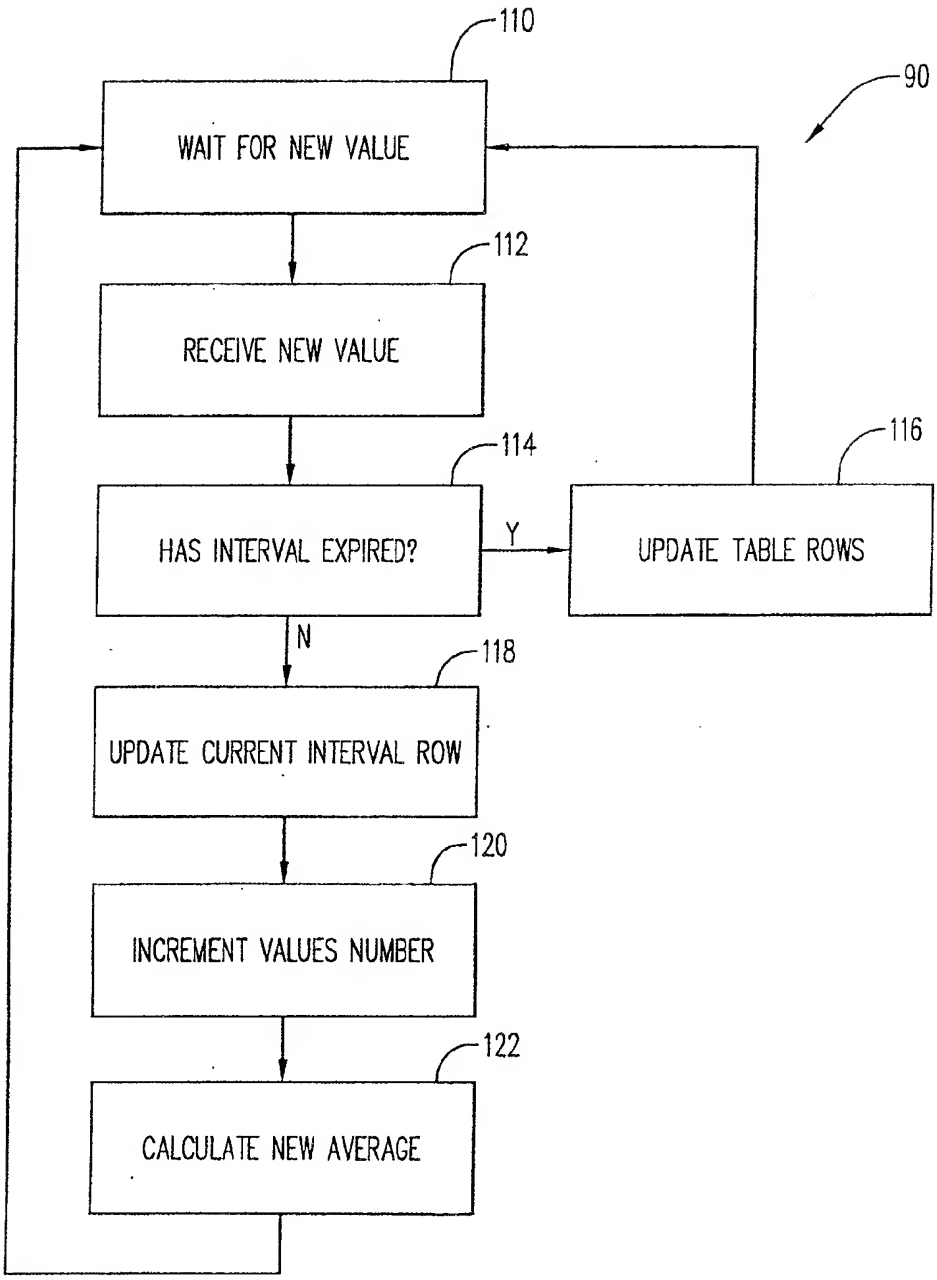
110
90
WAIT FOR NEW VALUE
112
RECEIVE NEW VALUE
114
116
HAS INTERVAL EXPIRED?
Y
UPDATE TABLE ROWS
N
118
UPDATE CURRENT INTERVAL ROW
120
INCREMENT VALUES NUMBER
122
CALCULATE NEW AVERAGE

FIG. 6

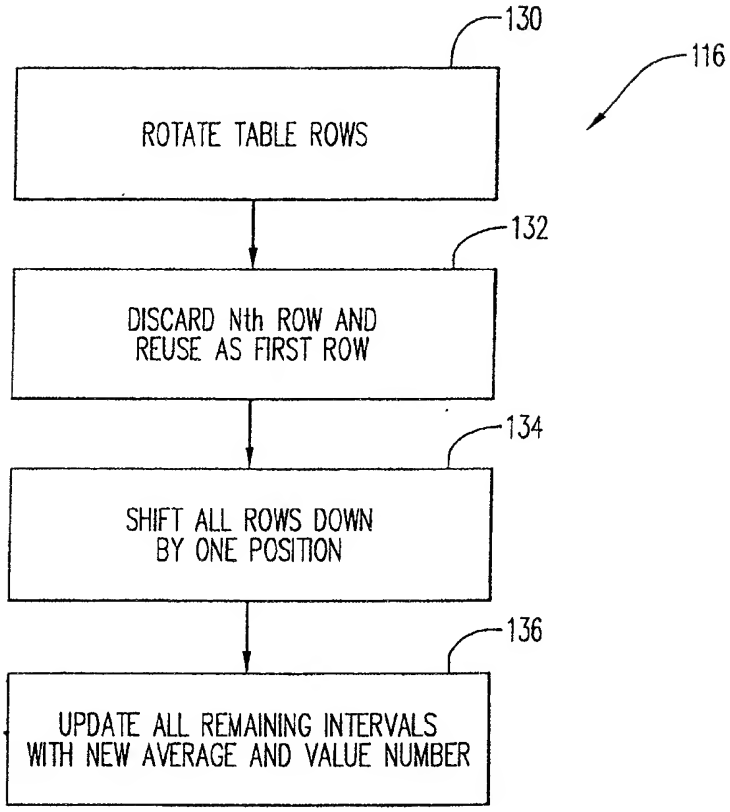
130
116
ROTATE TABLE ROWS
132
DISCARD Nth ROW AND
REUSE AS FIRST ROW
134
SHIFT ALL ROWS DOWN
BY ONE POSITION
136
UPDATE ALL REMAINING INTERVALS
WITH NEW AVERAGE AND VALUE NUMBER

FIG. 7